

The Hatch - Module Usage

Instructions on how to set up the module and the secret key were listed in the .pdf previously shared.

The module code is public and can be found here:

<https://create.roblox.com/store/asset/121738472086675/HatchDevModule>

Steps

1. Open your game in Roblox Studio
2. Insert a Script named HatchServer into ServerScriptService
3. Paste the following code:

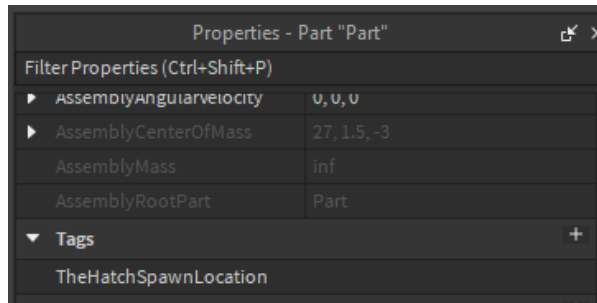
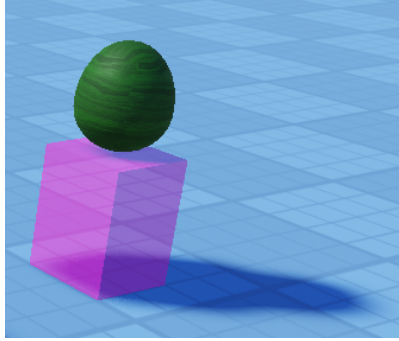
```
local DEV_MODULE_ID = 121738472086675
local HatchDevModule = require(DEV_MODULE_ID)
local config = {
    badgeld = 0,
    universeSecretKey = "",
    teleportToNextGameEnabled = true
}
HatchDevModule:Initialize(config)
HatchDevModule:EnableDefaultEggSpawning()
```

4. Replace badgeld with the badge you uploaded
5. Replace universeSecretKey with the value copied from The Form experience.
6. Play your game in Studio and verify that the egg spawns and you see a UI pop up when you touch it.

Module Behavior

Spawning

- By default, the eggs will spawn on any part tagged with 'TheHatchSpawnLocation' in the Workspace.
- We recommend making these parts have CanCollide = false and Transparency = 1 so you can't see or touch them.
- The eggs will spawn directly above the part, and will take the part's size into account. (example below)



- If you have multiple parts with this tag, a random part will be selected.
- If the part with the tag gets destroyed during the level, for example during a map switch, the old egg instance will get cleaned up and a new egg will automatically spawn, when a new part with that tag gets added to the Workspace.

Replication

- Each player will earn a different egg in your experience. For this reason, **the eggs are actually spawned client-side** using a RemoteEvent and **only exist on the client**.
- The server keeps track of which eggs should be spawned for each player, and validates the player is within a few studs of the egg location before awarding the egg, to prevent spoofing.

Module API Overview

Methods

HatchDevModule:Initialize(Config config)

Needs to be called once when your game starts up, with a config.

Config definition:

```
{
  -- The badge ID which you uploaded to grant players when they pick up the egg
  badgeId: number,
  -- The secret key used to determine which eggs each player gets in your experience.
  --You must use the value supplied from The Form.
  universeSecretKey: string,
```

```

-- Whether to allow teleports to the next game in the chain.
-- We highly recommend leaving this enabled.
teleportToNextGameEnabled: boolean
}

```

HatchDevModule:EnableDefaultEggSpawning()

Enables the default egg spawning algorithm, which auto-spawns eggs when each player joins.

HatchDevModule:IsEventLive() : boolean

Returns true if the event is currently live, based on the current `os.time()` being between July 2, 2025 6:00 PM GMT and July 12, 2025 8:00 PM GMT.

- Use this method to gate features that should only work when the event is running.
- This may be easier than pushing an update and shutting down all your servers.

HatchDevModule:SpawnEggForPlayerAtPositionAsync(Player player, Vector3 position) : boolean

- Spawns an egg for the given player directly above the given `Vector3` position on the map.
- The egg is automatically moved upwards by ~2.5 studs from the provided position, which is the approximate height of the eggs.
- If an egg is already spawned for that player, it is despawned and replaced by a new egg.
- Returns false if the module is not initialized, or the player has already unlocked the egg.
- Yields because this is dependent on checking if the badge is owned.

HatchDevModule:IsPlayerParticipating(Player player) : boolean

Returns whether the player is currently participating in the egg hunt.

- This is based on the presence of `TeleportData` supplied by The Hub.
- If a player joins your experience directly, this will be false.
- When testing in Studio, always returns true.

HatchDevModule:ShouldSpawnEggAsync(Player player) : boolean

Returns whether an egg should be spawned for a particular player.

Returns true only if both conditions are met:

- 1) Player joined with the `HatchPayload` data in their teleport (or via Studio)
- 2) Player does not have the badge yet.

HatchDevModule:GetEggForPlayer(Player player) : EggData

Returns an `EggData` table containing details about the egg that this specific player will earn in your experience.

Example of an `EggData`:

```

{
    -- String identifier for the egg, e.g. A1

```

```

    name: string

    -- MeshPart instance that represents the 3D egg model.
    -- Approximate dimensions are 2 studs wide, 2.5 studs tall, and 2 studs deep.
    model: MeshPart,

    -- 2D image URL for the egg.
    thumbnail: string
}

```

If the player is not participating in The Hatch, returns nil.

HatchDevModule:GrantEggToPlayerAsync(Player player)

Grants the badge to the player that corresponds to the egg.

Fires a remote event telling the client to pop-up a UI showing which egg was earned.

If config.teleportToNextGameEnabled is true, after we show the player which egg is earned, we give them another UI with 3 options:

- 1) Teleport back to The Hub
- 2) Teleport to the next game in the chain
- 3) Cancel (stay in the game)

HatchDevModule:GetNextTeleportDestination(Player player) : { universeld, placeId }

We call this function after earning an egg to see which place you should be sent to next.

Returns nil if you are at the end of the teleport chain, or if teleportToNextGameEnabled is false.